

# Conceptos de Interconexión de Redes

## Introducción a TCP/IP

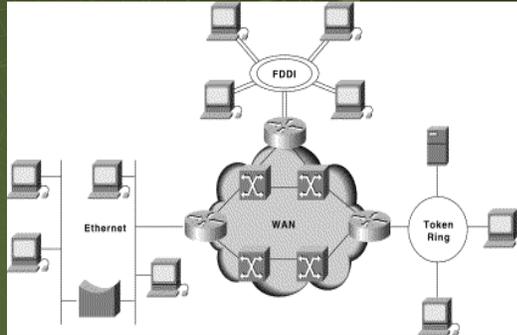
Carlos Vicente  
cvicente@ns.uoregon.edu

## Inter-redes

- ◆ Un poco de historia
  - ◆ Mainframes y terminales
  - ◆ PCs y redes locales
  - ◆ Redes de área extendida (X.25, ISDN, FR...)
  - ◆ Problema: Cada vez más, todas diferentes e incompatibles

# Inter-redes

## ◆ ¿Cómo interconectarlas?



# Inter-redes

- ◆ Se necesita una red más abstracta, a nivel superior, que esconda los detalles y diferencias de las diferentes redes físicas
- ◆ DARPA ya estaba trabajando en esto desde mediados de los 70!
- ◆ ARPANET: Una de las primeras redes de conmutación de paquetes
- ◆ ARPANET -> Universidades -> Industria
  - Se incluye IP en BSD unix
  - Primeros ISPs
- ◆ +230 Millones de máquinas conectadas (Enero 2004)
  - Ver: <http://www.isc.org/index.pl?ops/ds/>
- ◆ ¿Qué pasó?

# Diseño

- Muchos problemas que resolver:
  - Tecnologías cerradas y dispares
  - Pérdida de información
  - Control de flujo y congestión
  - Múltiples aplicaciones, un solo canal
  - Diferentes requerimientos de servicio
- Ley básica de ingeniería: Divide y vencerás
  - Dividir los problemas en grupos lógicos y jerárquicos
  - Esconder la complejidad, desacoplar
  - Facilitar la programación, prueba y mantenimiento

# Conceptos previos

- ◆ Servicios y redes orientados a conexión
  - Proveen garantías
    - ◆ Se pueden reservar recursos
  - Necesitan interacción entre los nodos
  - Implican un inicio y cierre de sesión
- ◆ No orientados a conexión
  - Sin garantías pero pueden ser más eficientes
- ◆ Cuando éstos se aplican a redes físicas, se suele hablar de:
  - Conmutación de circuitos (red telefónica)
  - Conmutación de paquetes (Internet)

## Tipos de envío

- ◆ Unicast
  - ◆ Uno a uno
- ◆ Broadcast
  - ◆ Uno a todos
- ◆ Multicast
  - ◆ Uno a varios
- ◆ Anycast
  - ◆ Uno a alguno

## Modelo de capas

- ◆ Modelo de referencia OSI
  - Sólo un modelo, no una arquitectura de red
  - Cada capa provee un servicio a la capa superior
  - Cada capa dialoga con su homóloga en el dispositivo remoto
  - Un protocolo es la implementación de la lógica de una capa
  - Uno o más protocolos por capa

Aplicación

Presentación

Sesión

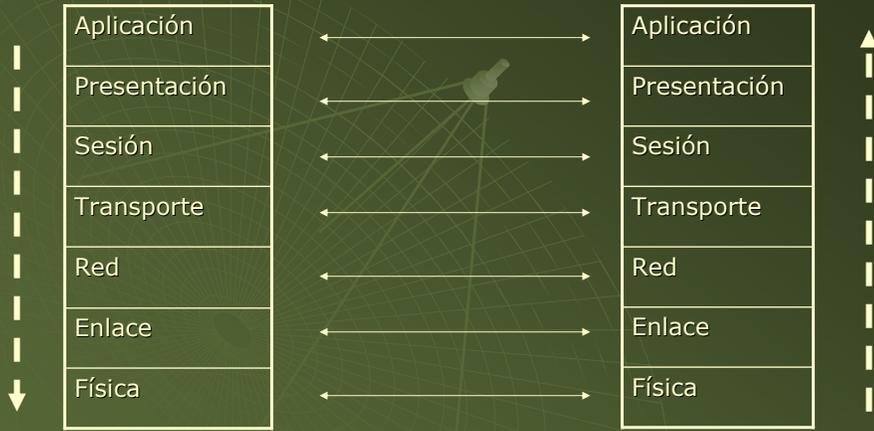
Transporte

Red

Enlace

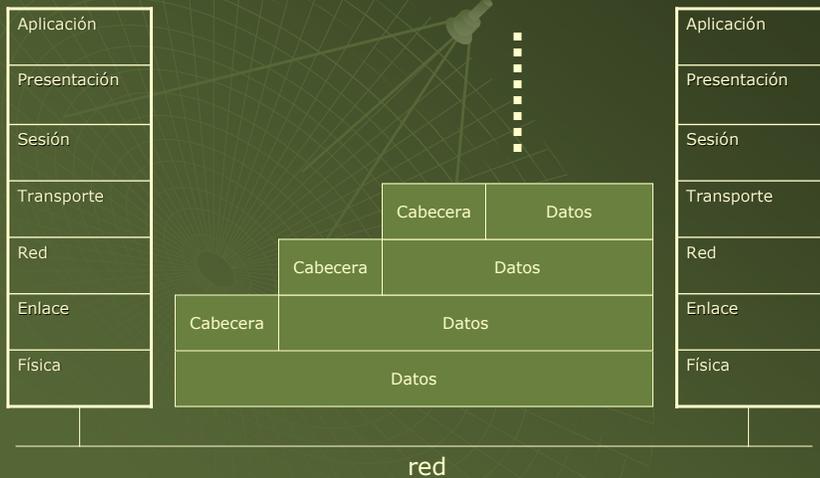
Física

# Modelo de Capas



# Modelo de Capas

## ◆ Encapsulación y cabeceras



# OSI vs. TCP/IP

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Física

Aplicación
Transporte
Red
Enlace
Física

HTTP, SMTP  
Telnet, POP

TCP/UDP

IP

Ej: Ethernet

- ◆ ARPANET empezó una década antes que OSI
- ◆ No necesitaban presentación ni sesión
- ◆ Estándar 'de facto'

## Capa 1: Física

- ◆ Implementada en hardware
- ◆ Codificación de canal
  - Representación de bits, voltajes, frecuencias, sincronización
    - ◆ Códigos Manchester, AMI, B8ZS...
- ◆ Define conectores físicos, distancias, cableado

## Capa 2: Enlace

- ◆ Encapsula los los paquetes en tramas para pasarlos al medio físico
- ◆ Reconstruye las tramas originales a partir de secuencias de bits y pasa los datos a la capa de red
- ◆ Provee
  - Direccionamiento (en el segmento de red local)
  - Detección de errores
  - Control de flujo

## Capa 3: Red

- ◆ Provee una red virtual global
  - Esconde los detalles de las redes físicas
  - Direccionamiento global:
    - ◆ Una dirección IP es suficiente para enviar hacia cualquier red en el mundo
    - ◆ Implica que hay que mapear las direcciones físicas con las IP
- ◆ Ofrece un servicio sin garantías (mejor esfuerzo)
  - Si se pierden o duplican paquetes, no le importa
  - Deja esa función a las capas superiores
- ◆ Determina si el destino es local o si lo debe enviar a un enrutador
- ◆ Provee funciones de control
  - ICMP
- ◆ Reenvía paquetes de salto en salto, de una red a la otra
  - El trayecto completo puede constar de muchos saltos

## Capa 4: Transporte

- ◆ Servicio con garantías (TCP)
  - Resuelve los problemas de:
    - ◆ Pérdida de paquetes
    - ◆ Duplicación
    - ◆ Desbordamiento (control de flujo)
- ◆ Sin garantías (UDP)
  - Mucho más simple
  - A veces no hace falta fiabilidad
- ◆ Provee multiplexión de aplicaciones
  - Concepto de 'puertos'

## Capa 5: Aplicación

- ◆ La más cercana al usuario
  - Define las funciones de clientes y servidores
- ◆ Utiliza los servicios de transporte
- ◆ Ej: HTTP (web), SMTP (mail), Telnet, FTP, DNS...

# Terminología

Aplicación
Transporte
Red
Enlace
Física

→ Segmento (TCP)

→ Datagrama (IP)

→ Trama, Frame (Ethernet)

- ◆ Nombres diferentes en cada capa

- ◆ No se sigue muy estrictamente. Suele hablarse indistintamente de 'paquete' en todas las capas.

# Tipos de enlaces

- ◆ Difusión (broadcast)
  - Ej: Ethernet
- ◆ Punto a punto
  - Ej. PPP, SLIP, HDLC
- ◆ NBMA (Non-broadcast Multi-Access)
  - Ej: Frame Relay, ATM

# Un vistazo a Ethernet

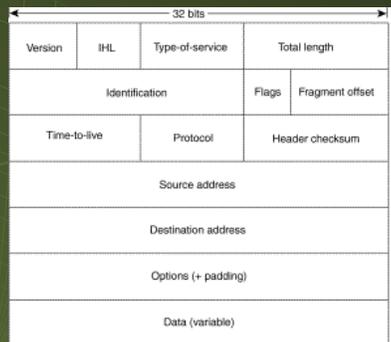
- ◆ Una red de difusión (broadcast)
  - Topologías
    - ◆ Bus (cable coaxial)
    - ◆ Estrella con repetidor
    - ◆ Estrella con conmutador
- ◆ ¿CSMA/CD?
- ◆ Razones para su éxito
  - Simplicidad
  - Costo
- ◆ De 10 Mbps a 10 Gbps

# Un vistazo a Ethernet

Preámbulo (8 bytes)	Destino (6)	Fuente (6)	Longitud (2)	Tipo (2)	Datos (46-1500)	FCS (4)
------------------------	----------------	---------------	-----------------	-------------	--------------------	------------

- ◆ Direcciones MAC:
  - Únicas y grabadas en el hardware de la tarjeta
    - ◆ Por eso también se llaman "direcciones físicas"
  - 6 bytes x 8 bits/byte = 48 bits
  - Suelen escribirse en hexadecimal
    - ◆ FE:D2:89:C4:4F:2E
- ◆ Tipo: 0x800 especifica que la parte de datos contiene un datagrama IP

# El datagrama IP



- ◆ Versión actual : 4
- ◆ El protocolo se refiere al que está siendo encapsulado (tcp, udp...)
- ◆ TTL se decremента con cada salto
- ◆ Hay fragmentación al pasar de un MTU mayor a uno menor

# La dirección IP

- ◆ Un número de 32 bits (4 bytes)

- Decimal:

128	223	254	10
-----	-----	-----	----

- Binaria:

10000000	11011111	11111110	00001010
----------	----------	----------	----------

- Hexadecimal:

80	DF	FE	0A
----	----	----	----

# La dirección IP

## ◆ Estructura

- Un sólo número, dos informaciones:
  - ◆ Dirección de la red (prefijo)
  - ◆ Dirección del nodo dentro de esa red

128	223	254	10
-----	-----	-----	----



## ◆ ¿Dónde está la división?

- Al principio era implícito (clases)
- Luego más flexible (máscaras)

# Esquema de clases (Classful)

Clase	Formato	Primeros bits	Rango	Bits por nodo
A	R.N.N.N	0	1.0.0.0 - 126.0.0.0	24
B	R.R.N.N	10	128.1.0.0 - 191.254.0.0	16
C	R.R.R.N	110	192.0.1.0 - 223.255.254.0	8
D	n/a	1110	224.0.0.0 - 239.255.255.255	n/a
E	n/a	1111	240.0.0.0 - 254.255.255.255	n/a

- ◆ Los límites red-nodo en la dirección son arbitrarios
- ◆ ¿Qué problema podemos prever?

# Revisión de base binaria

- ◆ En base 10 decimos:

- $1234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 = 1000 + 200 + 30 + 4$

- ◆ De la misma forma, en base 2:

- $1010 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 0 + 2 + 0 = 10$  decimal

- ◆ Suma lógica (AND):

- $1 + 1 = 1$
- $1 + 0 = 0$
- $0 + 1 = 0$
- $0 + 0 = 0$

# Potencias de 2 en un byte

- ◆ Conviene memorizar:

- $2^7 = 128$       1000 0000
- $2^6 = 64$       0100 0000
- $2^5 = 32$       0010 0000
- $2^4 = 16$       0001 0000
- $2^3 = 8$       0000 1000
- $2^2 = 4$       0000 0100
- $2^1 = 2$       0000 0010
- $2^0 = 1$       0000 0001

# Máscaras

- ◆ Solución: Otro número que especifique los límites

10000000	11011111	11111110	00001010	128.223.254.10
AND				
11111111	11111111	11111111	00000000	255.255.255.0
=				
10000000	11011111	11111110	00000000	128.223.254.0

Con esto se podían subdividir las redes A, B y C en subredes más pequeñas

# Notación de prefijo

- ◆ La máscara también se puede especificar como la cantidad de bits a 1:
  - 255.255.255.0 tiene 24 bits a 1
- ◆ Se agrega a la dirección IP con "/"
  - 128.223.254.10/24
- ◆ Hoy día se utilizan indistintamente las dos notaciones

## Direcciones especiales

- ◆ Todos los bits de nodo a 0: Representa la red
  - 128.223.254.0/24
- ◆ Todos los bits a 1: Broadcast local o limitado
  - 255.255.255.255
- ◆ Todos los bits de nodo a 1: Broadcast dirigido
  - 128.223.254.255
- ◆ Direcciones Loopback:
  - 127.0.0.0/8
    - ◆ Casi siempre se usa 127.0.0.1

## Más direcciones Especiales

- ◆ Direcciones privadas (RFC 1918)
  - 10.0.0.0 - 10.255.255.255 (10/8)
  - 172.16.0.0 - 172.31.255.255 (172.16/12)
  - 192.168.0.0 - 192.168.255.255 (192.168/16)
- ◆ ¿Cuál es la necesidad?

## Problemas con el esquema de clases

- ◆ No muy flexible
- ◆ Se perdían dos subredes en cada división
- ◆ En los 90's cambió el esquema (Classless):
  - ¡Las viejas clases A, B, C no tienen significado ninguno en el Internet de hoy!
    - ◆ CIDR (Classless Interdomain Routing):
      - Los routers ya no consideran A,B,C como /8, /16, /24
    - ◆ VLSM (Variable Length Subnet Masks)
      - Los routers no asumen que todas las subredes son del mismo tamaño
- ◆ ¿Es 128.223.254.0/24 una clase C?

## Subdividiendo una red

- ◆ Nos debemos preguntar:
  - Cuántas subredes queremos
  - Cuántos nodos tendrá cada subred
- ◆ Calculando en la cabeza
  - Cada bit más de máscara es el doble de subredes y la mitad de nodos (o viceversa)
  - Cada subred se puede sub-dividir a su vez
- ◆ Ejercicio:
  - Nos asignan 192.168.1.0/24
  - Tenemos:
    - ◆ 1 edificio con 121 nodos
    - ◆ 1 edificio con 50 nodos
    - ◆ 2 edificios con 25 nodos

## Algunos Trucos

- ◆ 255.255.255.192
    - A qué prefijo corresponde?
      - ◆  $256 - 192 = 64$  (hosts/subred)
      - ◆  $64 = 2^6$
      - ◆ Si tengo 6 bits de nodo, quedan 2 de red
- 11111111 11111111 11111111 11000000
- ◆  $3 \times 8 = 24, +2 = 26$
  - ◆ Prefijo es /26

## Datagrama IP

- ◆ Algunos campos interesantes
  - Type of Service (TOS)
    - ◆ retardo, fiabilidad, velocidad (voz vs. datos)
  - Identification, Flags, Fragment Offset
  - TTL

## Routers

- ◆ Dispositivos con interfaces en varias redes físicas
- ◆ Una dirección IP (y subred) por cada interfaz
- ◆ Deciden el trayecto de los paquetes basados en tablas de encaminamiento

## Envío

- ◆ En IP, distinguimos entre:
  - Envío directo:
    - ◆ La máquina envía a otra que está en su propia red física (Ej: mismo segmento Ethernet)
  - Envío indirecto:
    - ◆ El destino del paquete IP está fuera de la red física
      - Requiere la presencia de un router

## Envío y Re-envío

- ◆ Algoritmo de routing
  - Extraer la dirección IP de destino (D)
  - Si D encaja en alguna de las redes ( $R_i$ ) físicamente conectadas
    - ◆ Enviar directamente a D por la interfaz conectada a esa red
      - (Implica traducir la dirección IP a la física)
  - Sino, si la tabla contiene una ruta específica a D
    - ◆ Enviar el paquete al próximo salto especificado en la tabla

## Envío y Re-envío

- ◆ Algoritmo de Routing (Cont.)
  - Sino, si la tabla contiene una ruta a la red R que contiene a D
    - ◆ Enviar el paquete al próximo salto especificado en la tabla
  - Sino, si la tabla contiene una ruta por defecto
    - ◆ Enviar el paquete al router por defecto
  - Sino, notificar un error de routing

## Routers

- ◆ Mecanismo de un router:
  - Recibe un paquete en una interfaz
  - Determina si el paquete está dirigido a él
  - Decrementa el TTL
  - Compara la dirección destino con la tabla de encaminamiento
  - Envía el paquete al router del próximo salto (o a la máquina destino)

## Encaminamiento

- ◆ Cada decisión es un salto en la dirección de destino
  - El router no puede enviar a otro router que no esté en una de sus propias redes físicas
- ◆ Cada router tiene sus propias tablas
- ◆ Protocolos de routing: mantener estas tablas al día

# Tablas de Encaminamiento

- ◆ Se compara la dirección IP destino del paquete con las entradas en la tabla
- ◆ Determinar el próximo salto
  - Se asume que está físicamente conectado
- ◆ ¿Qué es la regla del *longest match*?

IP	Máscara	Gateway
192.168.5.0	255.255.255.0	192.168.1.1
192.168.5.0	255.255.255.192	192.168.1.2
0.0.0.0	0.0.0.0	192.168.0.1

# Tablas de Encaminamiento

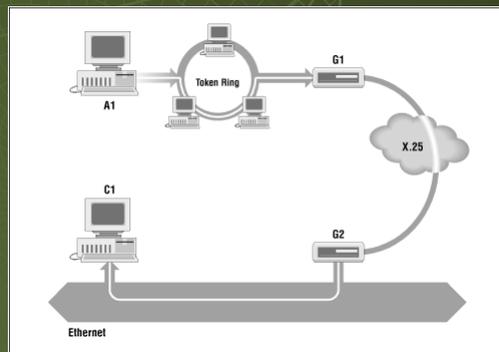
```
# netstat -nr
Kernel IP routing table
Destination    Gateway         Genmask         Flags   MSS Window  irtt Iface
128.223.60.0  0.0.0.0        255.255.254.0  u       0 0        0 eth0
127.0.0.0     0.0.0.0        255.0.0.0      u       0 0        0 lo
0.0.0.0       128.223.60.1  0.0.0.0        uG      0 0        0 eth0
```

# Routers y Encapsulación

1. Recibe una trama de capa 2
  2. Saca el paquete IP de ella
  3. Revisa el contenido del paquete IP
  4. Determina la interfaz siguiente
  5. Encapsula el paquete dentro de una trama del tipo correspondiente
- Las redes de entrada y salida pueden ser completamente diferentes:
    - ♦ Ejemplos:
      - De Ethernet a PPP
      - De Frame Relay a Ethernet

# Fragmentación

- ♦ Diferentes MTU en cada salto



# Traducción de direcciones

- ◆ Problema: Diferentes direcciones en capas de enlace y de red
  - Caso Ethernet:
    - ◆ Tengo este paquete IP para reenviar. ¿Qué dirección Ethernet tengo que poner en mi nueva trama?
  - Tres tipos de soluciones:
    - ◆ Derivar una de otra con cierta operación matemática
    - ◆ Incluir una dentro de la otra
    - ◆ Mantener una tabla dinámica

# ARP

- ◆ Mantiene tablas dinámicas

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>arp -a

Interface: 128.223.219.14 --- 0x2
Internet Address      Physical Address      Type
128.223.216.1         00-04-75-71-e5-64    dynamic
128.223.216.24        00-04-23-62-14-4f    dynamic
```
- ◆ Las entradas tienen un tiempo de vida limitado (¿Por qué?)
- ◆ Mecanismo:
  - A quiere enviar a B
  - Primero busca la IP de B en su tabla
  - Si no la tiene, pregunta
    - ◆ A: ¿Quién tiene 192.168.0.1?
  - Envía una trama a toda la red:
    - ◆ Utiliza `FF:FF:FF:FF:FF:FF` (todos los bits a 1)
    - ◆ Todos reciben la trama. Sólo el B responde

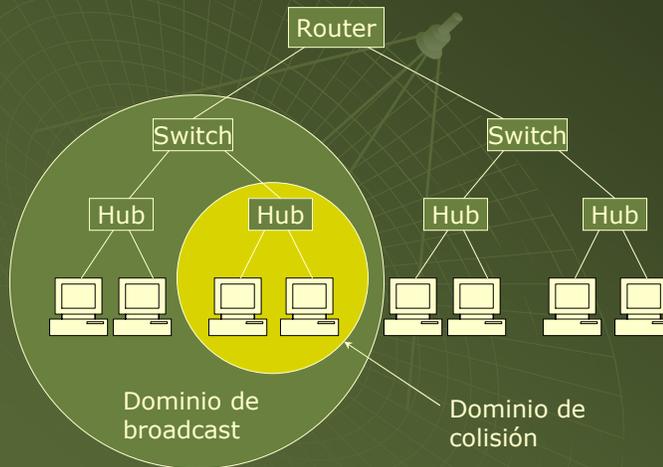
# ARP

- ◆ Algunas mejoras de eficiencia:
  - A quiere saber la MAC de B
  - B recibe la trama. Toma las direcciones MAC e IP de A y las incluye en su tabla
  - Luego B responde a A
  - Como la petición es broadcast, en principio todos los demás pueden incluir a A en su tabla.
- ◆ Pregunta: El paquete ARP viaja dentro de una trama Ethernet o un paquete IP?

# Dominios de Tráfico

- ◆ Dominio de colisión
- ◆ Dominio de broadcast
- ◆ Diferencias
  - Switches vs. Hubs
  - Routers vs. switches

# Dominios de tráfico



# Ventajas de las subredes

- ◆ Escalabilidad, eficiencia
  - Reducir los dominios de broadcast
    - ◆ Menos uso de CPU
    - ◆ Más espacio para tráfico legítimo -> más velocidad
- ◆ Facilitar la gestión
  - ◆ Ingeniería de tráfico
    - Implementación de políticas
  - ◆ Seguridad
    - Filtros de paquetes

# UDP

- ◆ User Datagram Protocol
  - Multiplexión de aplicaciones
    - ◆ Una dirección IP identifica una máquina
    - ◆ Los sistemas operativos son multitarea
    - ◆ Un puerto para cada servicio
  - ◆ Servicio no orientado a conexión
    - No ofrece ninguna garantía
      - ◆ Sin acuses de recibo
      - ◆ Sin re-transmisión
      - ◆ Sin control de flujo

# UDP

- ◆ Formato de UDP

Puerto Origen	Puerto Destino
Longitud	Checksum
Datos	
...	

# TCP

- ◆ Transmission Control Protocol
  - Orientado a conexión
    - ◆ Hay un acuerdo previo entre origen y destino
    - ◆ Hay un diálogo que va ajustando parámetros constantemente
  - Servicios:
    - ◆ Fiabilidad
      - Paquetes perdidos, duplicados, desordenados
    - ◆ Control de flujo
    - ◆ Multiplexión de aplicaciones

## TCP: Conceptos

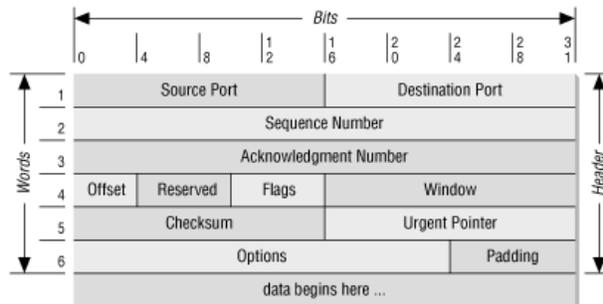
- ◆ PAR: Positive Acknowledgment with Retransmission
  - Envío un segmento e inicio un timer
  - Espero una confirmación antes de enviar el siguiente
  - Envío el mismo segmento de nuevo si el tiempo expira sin recibir confirmación
- ◆ ¿Segmentos duplicados? ¿Cómo?
  - Un retraso en la red produce una retransmisión y el mismo segmento llega dos veces

# TCP: Ventana deslizante

- ◆ Esperar confirmación por cada paquete no es muy eficiente
  - Tamaño de ventana = 1
- ◆ También resuelve el problema de control de la congestión
  - El tamaño de la ventana se ajusta dinámicamente

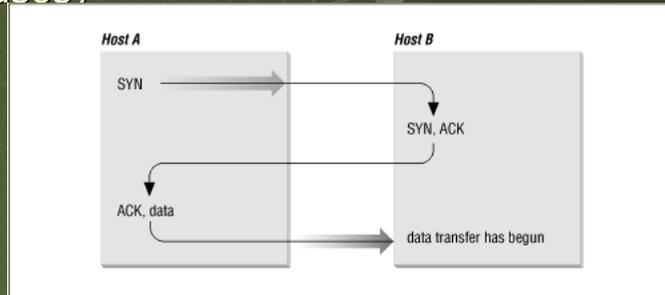


# Formato de TCP



## TCP: Inicio de Sesión

- ◆ Three-way handshake (saludo en tres pasos)



## ¿TCP o UDP?

- ◆ Cuándo tiene sentido uno u otro
  - FTP
  - DNS
  - SNMP
  - Voz sobre IP (H.323, SIP)
  - Multicast

# ICMP

- ◆ Internet Control Message Protocol
  - Viaja sobre IP, pero no pertenece a la capa de transporte
  - Funciones:
    - ◆ Notificar errores
    - ◆ Control de Flujo
    - ◆ Redirección

# ICMP

- ◆ Algunos tipos y códigos más usados

Tipo	Código	Descripción
0	0	Echo Reply
3	0	Destination Network unreachable
3	1	Destination Host Unreachable
3	2	Destination Protocol Unreachable
3	3	Destination Port Unreachable
8	0	Echo Request
11	0	TTL expired

# ICMP: Aplicaciones

## ◆ Ping

```
# ping www.uoregon.edu
PING darkwing.uoregon.edu (128.223.142.13) from 128.223.60.27 : 56(84) bytes of data.
64 bytes from darkwing.uoregon.edu (128.223.142.13): icmp_seq=1 ttl=254 time=0.229 ms
64 bytes from darkwing.uoregon.edu (128.223.142.13): icmp_seq=2 ttl=254 time=0.254 ms
64 bytes from darkwing.uoregon.edu (128.223.142.13): icmp_seq=3 ttl=254 time=0.226 ms
64 bytes from darkwing.uoregon.edu (128.223.142.13): icmp_seq=4 ttl=254 time=0.232 ms
64 bytes from darkwing.uoregon.edu (128.223.142.13): icmp_seq=5 ttl=254 time=0.222 ms
```

# ICMP: Aplicaciones

## ◆ Traceroute

```
# traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 66.102.9.99
traceroute to www.google.akadns.net (66.102.9.99), 30 hops max, 38 byte packets
 1  ge-4-6.uonet2-gw.uoregon.edu (128.223.60.3)  0.310 ms  0.236 ms  0.193 ms
 2  0.ge-0-0-0.uonet8-gw.uoregon.edu (128.223.2.8)  0.324 ms  0.331 ms  0.294 ms
 3  eugn-car1-gw.nero.net (207.98.66.11)  0.363 ms  0.296 ms  0.416 ms
 4  eugn-core2-gw.nero.net (207.98.64.169)  0.672 ms  1.029 ms  0.601 ms
 5  ptck-core2-gw.nero.net (207.98.64.2)  2.911 ms  2.994 ms  2.930 ms
 6  ptck-core1-gw.nero.net (207.98.64.137)  3.255 ms  2.874 ms  2.923 ms
 7  so-6-1.hsa2.Seattle1.Level3.net (63.211.200.245)  6.521 ms  6.153 ms  6.322 ms
 8  ge-6-1-1.mp2.Seattle1.Level3.net (209.247.9.85)  6.619 ms  6.565 ms  6.335 ms
 9  so-0-0-0.bbr2.NewYork1.Level3.net (64.159.0.238)  86.194 ms  86.239 ms  86.580 ms
10  so-2-0-0.mp2.London1.Level3.net (212.187.128.154)  147.899 ms  147.968 ms  149.461
    ms
11  so-3-0-0.mp2.Amsterdam1.Level3.net (212.187.128.13)  155.019 ms  155.738 ms
    155.406 ms
12  ge-11-2.ipcolo2.Amsterdam1.Level3.net (213.244.165.116)  157.499 ms  155.627 ms
    155.857 ms
13  212.72.44.66 (212.72.44.66)  156.319 ms  156.168 ms  156.142 ms
```

# Traceroute: Funcionamiento

```
# traceroute 128.223.142.13
traceroute to 128.223.142.13 (128.223.142.13), 30 hops max, 38 byte packets
 1 ge-4-6.uonet2-gw.uoregon.edu (128.223.60.3)  0.282 ms  0.206 ms  0.186 ms
 2 darkwing (128.223.142.13)  0.266 ms  0.197 ms  0.209 ms

(simultáneamente)

# topdump -lnv host 128.223.142.13 of icmp
topdump: listening on eth0
128.223.60.27.33962 > 128.223.142.13.33435: udp 10 [ttl 1] (id 12001, len 38)
128.223.60.3 > 128.223.60.27: icmp: time exceeded in-transit [tos 0xc0] (ttl 255, id 64235, len 56)
128.223.60.27.33962 > 128.223.142.13.33436: udp 10 [ttl 1] (id 12002, len 38)
128.223.60.3 > 128.223.60.27: icmp: time exceeded in-transit [tos 0xc0] (ttl 255, id 64236, len 56)
128.223.60.27.33962 > 128.223.142.13.33437: udp 10 [ttl 1] (id 12003, len 38)
128.223.60.3 > 128.223.60.27: icmp: time exceeded in-transit [tos 0xc0] (ttl 255, id 64237, len 56)
128.223.60.27.33962 > 128.223.142.13.33438: udp 10 (ttl 2, id 12004, len 38)
128.223.142.13 > 128.223.60.27: icmp: 128.223.142.13 udp port 33438 unreachable (DF) (ttl 254, id 14809, len 66)
128.223.60.27.33962 > 128.223.142.13.33439: udp 10 (ttl 1, id 12005, len 38)
128.223.142.13 > 128.223.60.27: icmp: 128.223.142.13 udp port 33439 unreachable (DF) (ttl 254, id 14810, len 66)
128.223.60.27.33962 > 128.223.142.13.33440: udp 10 (ttl 2, id 12006, len 38)
128.223.142.13 > 128.223.60.27: icmp: 128.223.142.13 udp port 33440 unreachable (DF) (ttl 254, id 14811, len 66)
```

# Telnet a puertos conocidos

```
# telnet www.uoregon.edu 80
Trying 128.223.142.13...
Connected to www.uoregon.edu.
Escape character is '^J'.
GET /
<html>
<head>
<title>University of Oregon Home Page</title><link rel="SHORTCUT ICON"
href="/favicon.ico">

# telnet darkwing.uoregon.edu 25
Trying 128.223.142.13...
Connected to darkwing.uoregon.edu.
Escape character is '^J'.
220 darkwing.uoregon.edu ESMTSP Sendmail 8.12.11/8.12.11; Mon, 23 Feb 2004 15:52:28 -
0800 (PST)
vrfy cvicente
250 2.1.5 Carlos Vicente <cvicente@darkwing.uoregon.edu>
quit
221 2.0.0 darkwing.uoregon.edu closing connection
Connection closed by foreign host.
```

## Ejercicios en clase

- ◆ ¿A qué subred pertenece 172.17.21.10/22?Cuál es la dirección de broadcast correspondiente?

## Ejercicios en Clase

- ◆ Solución:
  - ◆ /22 significa que en el tercer octeto tenemos 6 bits de subred
  - ◆  $2^6 = 64$
  - ◆ Si tenemos 256 posibles números para subredes, y sólo 64 válidas
    - $256/64 = 4$
    - Las subredes irán de 4 en 4, no?
    - 172.17.0.0 (172.17.0.1 - 172.17.3.255)
    - 172.17.4.0 (172.17.4.1 - 172.17.7.255)
    - 172.17.8.0 (172.17.8.1 - 172.17.11.255)
  - ◆ Nuestra dirección cae dentro de 172.17.0.20
  - ◆ La dirección broadcast es la última en ese rango
    - 172.17.23.255

## Ejercicios en Clase

- ◆ ¿Qué pasaría si configuramos un nodo con una máscara incorrecta? Por ejemplo, 255.255.254.0 cuando el router y los demás en la subred tienen 255.255.255.0?

## Ejercicios en Clase

- ◆ **Solución:**
  - No podrá enviar al rango IP contiguo
  - Por ejemplo:
    - ◆ La red es 192.168.0.0/24
    - ◆ El nodo está configurado con 192.168.0.10/23
    - ◆ El nodo piensa que su rango local es:
      - 192.168.0.1 – 192.168.1.254
    - ◆ Cuando envíe a nodos entre
      - 192.168.1.1 – 192.168.1.254
      - Enviará peticiones ARP que nunca serán respondidas

## Más información

- ◆ *TCP/IP Illustrated*. Richard Stevens. Addison-Wesley
- ◆ *Internetworking with TCP/IP*. Douglas Comer. Prentice-Hall
- ◆ *Cisco Internetworking Basics*  
[http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\\_doc/introint.htm](http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/introint.htm)
- ◆ *TCP/IP Network Administration*. Craig Hunt O'reilly & Associates.
- ◆ Requests for Comments (RFCs)  
[www.ietf.org](http://www.ietf.org)